## REMARKS

This is a response to the final Office Action mailed on July 3, 2008.

### Amended and new claims

Claims 1, 2, 4-7, 9, 11, 21, 24, 27, 29, 30, 32, 34, 36-40, 42, 45, 48 and 49 are amended. Claims 50-52 are new. Claims 20, 33, 43 and 44 are cancelled. No new matter is presented. Example support for the claims is as follows: claims 1, 21, 27, 29, 32, 34, 36, 38, 39 and 45 (Fig. 4 and 5, p.6, lines 14-20, p.16, lines 23 and 24), claims 2 and 30 (a period is added at the end of the claim), claims 5 and 6 (amended based on claim 4), claims 7, 9, 48 and 49 (amended for consistency with claim 1), claims 11 and 24 (p.24, lines 21-26), claim 37 (claim 5), claim 40 (amended for consistency with claim 39), claim 42 (claim 43), claim 50 (Fig. 2, p.6, lines 14-20, p.23, lines 12-17), claim 51 (p.3, lines 5 and 6, p.4, lines 18-20) and claim 52 (p.4, lines 6-9, Fig. 4).

### Rejection under 35 U.S.C. §112

Claims 1-11, 29-40, 48 and 49 have been rejected under 35 U.S.C. §112, second paragraph. Applicants submit that the claims as amended are clearly in compliance with 35 U.S.C. §112, second paragraph. It is clarified that existing object code is edited to provide modified object code, and the modified object code is executed. Withdrawal of the rejection is therefore respectfully requested.

### Rejection under 35 U.S.C. §102(b)

Claims 1-6, 8, 9, 12-18, 20-22, 25-27, 29-34, 36-39 and 41-46 have been rejected under 35 U.S.C. §102(b) as being anticipated by Chow (US 6,131,187). Applicants respectfully traverse the rejection.

Claim 1 sets forth accessing existing object code of a first application, where the existing object code is not configured to provide access to a result (such as a return value or exception) of a first method by an additional method for which such access is desired. The method further includes editing the existing object code to provide modified object code which provides access to the result by the additional method when the first method is executed. The additional method can be part of the same application which includes the first method, or part of another application, for instance. The method further includes executing the modified object code. In contrast, Chow discloses a computer system that translates exception handling semantics for a method from object code, such as Java byte code, to compilable high-level code such as C statements (col. 3, lines 50-61; col. 5, lines 40-43). Chow thus needs to recompile the code, which includes the C statement source code. This is a problem that Applicants seek to avoid (see Applicants' specification, p.4, lines 6 and 7).

Further, Chow is not concerned with modifying existing object code which is not configured to provide access to a result (such as a return value or exception) of a first method by an additional method for which such access is desired. This is desirable, for instance, when the additional method is used for debugging (specification, p.3, line 27), and the original object code is not configured to provide such access.

Instead, Chow simply seeks to optimize the translation of high level code by a compiler. There is no mention that an additional method which does not have access to a result is able to gain access to the result. In particular, Chow is concerned with the fact that exception handling semantics are not easily optimized by the compiler. This is due to artificial exception ranges kept in the exception table of a method which preclude the final natively compiled instructions from being rescheduled for optimal performance (Chow, col. 1, lines 43-63). However, this is a different problem that modifying existing object code which is not configured to provide access to a result of a first method by an additional method for which such access is desired.

Accordingly, claim 1 and its dependent claims are clearly patentable.

Regarding claim 2, col. 4, lines 40-66 of Chow refer to pertinent information required for each entry within a bytecode information array. However, there is no disclosure or suggestion that object code is modified to allow an additional method to have access to this information.

Regarding claim 3, similarly, while a reference to an exception is know *per se*, there is no disclosure or suggestion of modifying object code to allow an additional method to have access to the exception.

Regarding claim 4, Chow provides no disclosure or suggestion at least of adding code that prepares an operand stack for an invocation of an additional method, and adding code that invokes the additional method, including providing a result to the additional method, as part of editing object code to allow access to the result by the additional method.

Claims 5 and 6 which depend on claim 4 similarly distinguish over Chow.

Chow similarly fails to disclose or suggest the steps of claim 9 which are performed as part of editing of object code as claimed.

Regarding claim 12, Chow merely indicates how an interpreter typically monitors an operand stack. It is unclear how the Examiner believes this passage discloses the features of claim 12. However, even using the Examiner's explanation that the "result" can be equated to an exception produced by a "first method" in Chow, Chow does not disclose that this "result" is stored "from an operand stack." Nor does Chow disclose that the "operand stack" is prepared for "an invocation of a second method" or that the "result" is provided to the "second method" when the "second method" is invoked. Therefore, claim 12 and its dependent claims are patentable over Chow.

For example, claim 21 sets forth modifying byte code for a first method, where the byte code is not configured to provide a result of the first method to a second method when the second method is invoked. The modifying includes configuring the byte code to perform steps of storing, preparing, invoking and resetting as set forth in claim 12. In contrast, Chow is not concerned with configuring byte code to provide a result of a first method to a second method, where the byte code is not otherwise configured to perform this function.

Rejection under 35 U.S.C. §103(a)

Claims 7, 10, 11, 19, 23, 24, 28, 35, 40, 47 and 48 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Chow in view of Kukol (US 5,628,016).

Kukol discloses registering exception handlers with a computer's operating system so that when an exception occurs, an exception handler can easily be found using the stored exception registration records. However, this is not performed as part of editing object code to allow an additional method to access a result of a first method.

Regarding claim 11, while Kukol mentions a try block 905 and a finally block 908, there is no disclosure or suggestion of editing existing object code by adding Try-Finally functionality which is not included in the existing object code, as claimed.

Regarding claims 48 and 49, Kukol, taken alone or in combination with Chow, fails to disclose or suggest adding a tracer or a timer for a first method as part of editing object code to enable an additional method to access a result of the first method.

New claim 50 is patentable as the cited references are not concerned with enabling an additional method which is part of a monitoring program to access a result of a method of a monitored application.

Similarly, new claim 51 is patentable as the cited references are not concerned with enabling an additional method which is part of an application to access a result of a method of the same application.

New claim 52 is patentable as the cited references are not concerned with providing modified object code from existing object code without recompiling the existing object code. In contrast, Chow translates exception handling semantics for a method from object code, such as Java byte code, to compilable high-level code such as C statement (col. 3, lines 50-61; col. 5, lines 40-43), and thus needs to recompile the code to run it.

Conclusion

In view of the above, each of the presently pending claims is believed to be in condition for immediate allowance. The Examiner is therefore respectfully requested to pass this application on to an early issue. Should further questions remain, the Examiner is invited to contact the undersigned agent by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: August 14, 2008

By: /Ralph F. Hoppin /
Ralph F. Hoppin
Reg. No. 38,494

VIERRA MAGEN MARCUS & DENIRO LLP
575 Market Street, Suite 2500
San Francisco, California 94105-4206
Telephone: (415) 369-9660, x214
Facsimile: (415) 369-9665